Flexible FPGA Based Digital IC Test Development Education Laboratory Design and Application

Ian Grout
Department of Electronic and Computer Engineering
University of Limerick
Limerick, Ireland
Ian.Grout@ul.ie

***Abstract*—In this paper, a teaching aid for digital integrated circuit (IC) test development engineering education using the field programmable gate array (FPGA) is presented and discussed. The set-up allows for different digital IC test development scenarios to be configured within the FPGA. For analysis and evaluation purposes, embedded machine learning functions could also include automated reporting of the system use. A case study design, using a Xilinx Artix-7 FPGA, incorporating a circuit test set-up for an example digital IC based an implementation of the IEEE Std 1149.1 "IEEE Standard for Test Access Port and Boundary-Scan Architecture" is embedded within the FPGA. This is accessed by the user via a serial port connection. This case study design allows a user to investigate and implement test program development scenarios within a suitable education environment.**

Keywords—test engineering, machine learning, JTAG, VHDL, FPGA

I. Introduction

Electronic systems are used in a range of everyday applications and utilize a range of possible hardware and software implementation approaches to create the required functionality. To support system development and functionality modification in the application environment, many solutions are based on the software programmed processor such as the microprocessor (µP). Here, an application program is written to control a fixed architecture electronic circuit, and the circuit behaviour is modified by changing the software program. However, in general, adaptable software and hardware functions can be developed, and the alternative to the software programmed processor is the programmable logic device (PLD) [1]. Within the PLD, the configurable (programmable) hardware is configured to implement the required hardware functions. Applications range from industrial, home and social, to education.

The application area considered in this paper is higher education, specifically digital integrated circuit (IC) circuit and system test [2], [3], [4] development teaching and learning. For laboratory experiment development, the system developed is based on the field programmable gate array (FPGA). The FPGA is suited for complex digital operations such as digital signal processing (DSP), allows for hardware and software components to be designed, and modern FPGA architectures support evolving DSP requirements such as machine learning (ML) and deep learning (DL). This gives the designer a range of possible options to realize a set of circuit requirements. The system described here will have different possible implementation approaches, depending on the user requirements. The system, forming an engineering teaching laboratory resource, can be developed within the FPGA utilizing the ability of the FPGA to embed hardware modules as well as software programmed processors. This laboratory arrangement is depicted in Fig. 1 where two different scenarios are identified, both physically *local* and *remote*:

1. **Top (*local*)**: The experiment hardware is physically local to the user. The user interacts with the FPGA, and possibly additional external circuitry, via a computer software application.
2. **Bottom (*remote*)**: The experiment is remotely located from the user. The user interacts with the FPGA via a network or internet connection. The FPGA forms a remote laboratory [5] and allows multiple users to access the hardware resource.

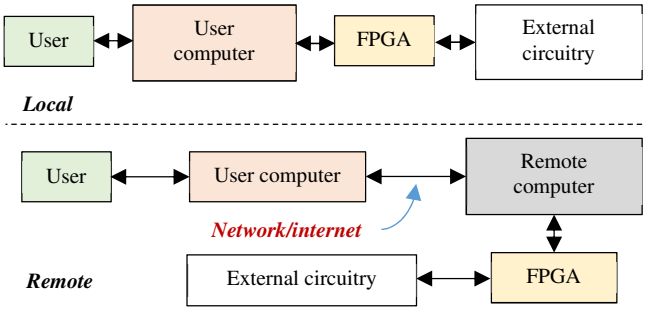


Fig. 1. System set-up to develop and evaluate

In this paper, a programmable test development engineering laboratory based on a Xilinx Artix-7 FPGA [6] is presented and discussed. The purpose of this FPGA based approach is to create a programmable platform that can be used to emulate different IC *design* and *test* scenarios to meet the requirements of a specific group of learners. This can be seen as one possible approach to creating a laboratory learning experience, as identified in Table I. Each approach would have its own set of advantages and disadvantages. The approach developed here would have advantage in that it could provide a single platform, fixed position or mobile, that can be quickly configured in a time and cost-effective manner. It could also be used to support at-presence and remote learning.

TABLE I. Approaches to Undertaking Laboratory Experiments in IC Test Engineering

Software simulation model based	
Functional simulation: circuit, system level; analog and digital.	Automatic test pattern generation (ATPG) tool.
Fault simulation: circuit, system level; analog and digital.	Software emulation of the hardware circuit using a software program on a user computer that models the functionality of the fault-free circuit hardware. May also include programmable faults.
Hardware based	
Actual physical circuit.	Hardware emulation of the fault-free circuit. May also include programmable faults.

This paper is arranged as follows. Section II will introduce the need for IC test development engineering education and a *circuit under test* (*CUT*) case study design based on an IEEE Std 1149.1 "IEEE Standard for Test Access Port and Boundary-Scan Architecture" [7]. This is an embedded hardware macro that will be a peripheral connected to an embedded Xilinx MicroBlaze [8] processor within the FPGA. The design implemented within the FPGA is therefore a mixed hardware and software design. Section III will consider the inclusion of machine learning concepts into the FPGA for data collection and analysis based on the use of the MicroBlaze processor with attached memory. Section IV will discuss the operation of the case study design implementation, and section V will conclude the paper.

II. Digital IC Test Development Education

Within the microelectronics industry, there are key engineering disciplines required to come together in cohesive teams to realise a working IC. Specifically considered here are the disciplines of *design*, *fabrication*, and *test*. The engineers within each discipline are required to interact with each other as a team in various and complex ways during the development of an IC in order to realize an IC design that meets all requirements. **Design engineers** are required to create a circuit design that meets the defined specifications. **Fabrication (process) engineers** are required to fabricate the design to the appropriate quality level. **Test engineers** are required to develop and implement suitable test programs that will verify the operation of the fabricated designs [1], [2], [3]. These engineering disciplines are required to closely interact so that, see Fig. 2, the design that has been created can be fabricated with the selected fabrication process, and the fabricated design can be tested to ensure a suitable circuit test coverage is achieved and to enable reliability assessment data collection. Underlying these operations are the needs to create and deploy new IC designs in a timely and cost-effective manner.

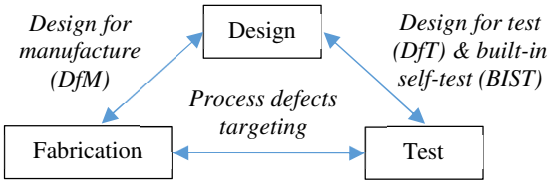


Fig. 2. Design, fabrication, and test interactions

Test engineers work in different roles with the final goal to develop a production test arrangement (both hardware and software) for a specific *device under test* (*DUT*) or *CUT*. Initial work involves the creation of a test specification, the development of suitable test hardware for wafer and packaged device testing, and software test programs to run on the tester hardware. These test programs can be for initial device prototyping actions that include design debugging, characterization, and evaluation or for final production testing (high-volume, high-speed, automated testing). With the increased complexity of ICs (more circuitry in a single device), lower geometry fabrication processes (more circuitry per mm^2), and higher operating frequencies, the design, fabrication, and test challenges are increasing. One important aspect for test is the inclusion within the design of embedded test circuitry that enables internal parts of the circuit not accessible at the primary inputs and outputs (the external connections) to be accessed. Such an approach for including embedded test circuitry is referred to as *design for test/design for testability* (DfT) and *built-in self-test* (BIST). The term DfT is generally referred to in relation to providing an external tester with test access to internal parts of a design, whilst BIST is generally referred to the inclusion of embedded test circuitry that enables the design to test parts of itself independent of an external tester. For digital IC testing, an important DfT technique that is used to support IC testing is the IEEE Std 1149.1 "IEEE Standard for Test Access Port and Boundary-Scan Architecture" [7]. The latest version of this IEEE standard was released in 2013. This standard was developed from the work of JTAG (Joint Test Action Group) in the 1980s to support digital IC testing once the IC had been soldered to its application printed circuit board (PCB). It can therefore be used to test the IC during the production process and once the IC has been mounted on a PCB. The idea is to provide test access via the device input and output pins, and to control the device to operate in a *normal operating mode* (mission mode) and one or more *test modes*. An example, the case study design considered in this paper, is shown in Fig. 3. This design implements a specific case use of the standard for teaching purposes where the IC design core is a 4-bit full-adder design. The design is small, when compared to the expected size and complexities of real-world applications but is suitably detailed to enable the principles to be learnt in a specified time period. The target audience for this work is higher education teaching of IC test development concepts to students of electronic and computer engineering disciplines. The circuit arrangement is a useful case study for a teaching and learning environment, and different aspects of design and test can be investigated in different learning scenarios. For example, digital IC design, DfT/BIST, IEEE standard implementation, and test program development that targets functional operation or fault conditions, could be envisaged with only a change to the hardware module within the FPGA that implements the *DUT*/*CUT* digital logic. Therefore, the programmability and hardware nature of the FPGA makes it an ideal platform for developing a range of laboratory experiment resources.

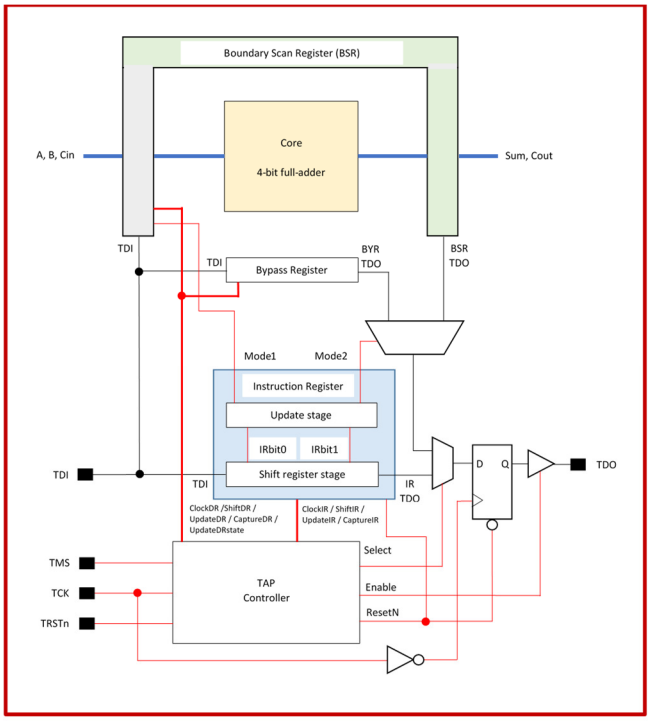


Fig. 3. Boundary scan compliant IC case study design

With this set-up, the design core is surrounded by a boundary scan register (*BSR*) that provides the core I/O (input/output) with both *normal mode* signals and *test mode* signals. This allows the testing of the IC core as well as the IC I/O and external PCB interconnects. The heart of the test hardware is the *TAP* (test access port) controller that is a 16-state FSM (finite state machine). The *TAP* controller is used to control the operation of the data registers (the *BSR* and the bypass register (*BYR*)), and the instruction register (*IR*). Serial test data can be loaded using the *TDI* (test data input) pin, and serial test results can be read using the *TDO* (test data output) pin. The *TAP* controller is operated using the *TMS* (test mode select), *TCK* (test clock), and optional *TRSTn* (test reset) inputs. Testing of a circuit would be aimed at one of two approaches, functional testing or structural testing. Whilst both approaches access the same circuit inputs with test vectors, and monitor the same circuit outputs to determine correct or incorrect circuit operation, they are looking at different aspects of circuit operation. Functional testing aims to determine whether the circuit functionality is correct, whereas structural testing is looking specifically at whether fabrication defects, if they exist, cause faulty circuit operation. Structural testing is undertaken to detect specific types of fabrication defects, and these defects are modelled as behaviour that can be inserted into a model of the *CUT*. The key faults considered in digital logic are the *stuck-at fault* (SAF), *bridging fault* (BF), *delay fault* (DF), and I_{DDQ} fault. The behaviour of these faults can be modelled as circuit elements for analysis. Structural testing is undertaken in a production test environment to reduce the test time when compared to a functional test approach.

III. Embedded Machine Learning Opporunities

Machine learning and deep learning algorithms are embedded in many electronic systems today to provide for enhanced functionality when it comes to the need for data analysis. The use of ML and DL will depend on the beneficial role, if any, that ML and DL can provide in the application. For an IC test application, two areas could potentially benefit from the use of ML and DL. These are:

1. Test data processing in a test environment for data analysis [9], [10], [11].
2. Learning analytics in an education laboratory environment.

Different ML and DL algorithms have been developed for different purposes. ML falls under different categories as unsupervised learning, supervised learning, semi-supervised learning, reinforcement learning, and DL. A suitable ML algorithm, or combination of algorithms, can be chosen to provide information using the data collected by the system. In the context of this work, the uses of ML include the ability to analyze the user input stimulus and results obtained to automatically identify individual user and class actions. The experiment itself can collate user and class data based on who has used the experiment, and how. For maximum benefit however, the FPGA requires processor and program/data memory resources to achieve useful functionality. It would also require access to non-volatile memory to prevent data loss if the power supply was to be removed. For this case study, the hardware utilization and the software program development are discussed in section IV. From the utilization statistics, the ability to include additional functions for ML operations exists, and so the FPGA can provide for a flexible and expandable platform for developing a laboratory experiment unit with embedded ML capabilities. An example user action would be to create and upload stimulus data representing the binary values to be applied to the experiment inputs. As part of the data upload procedure, the user would also need to provide a user ID. The FPGA would register the user ID, the time that the user ID was received and an experiment ID. The test run time statistics would be collated automatically and stored in memory for all users.

IV. Case Study

The case study system design presented here is based on a software programmed (C/C++) MicroBlaze 32-bit RISC (reduced instruction set computer) processor embedded within the Artix-7 FPGA. The *CUT* is included as a peripheral block connected to the MicroBlaze processor core using the Advanced eXtensible Interface (AXI) protocol. The processor provides the following functions:

1. *Control signal and test data transfer* between the connected computer and the FPGA via a UART (universal asynchronous receiver transmitter).
2. Test stimulus application and results capture *storage*.
3. *Machine learning* algorithms for test results data *analysis*. Analysis results transmission.
4. *Interfacing* to the embedded internal *CUT*.
5. *Interfacing* to an external *CUT*.

The hardware was developed using Xilinx Vivado 2019.2 [12] and the MicroBlaze programmed in C [13] using Xilinx Vitis IDE 2019.2 [14]. The block design in Vivado 2019.2 is shown in Fig. 4, and this shows the processor connected to peripheral circuits for both normal processor operation and the *CUT* implementation. The *CUT*, as shown diagrammatically in Fig. 3, is accessed by writing and reading 32-bit words to and from the address where the peripheral is in processor memory. In this set-up, the FPGA is mounted to the Digilent Arty-35T development board [15] and operates on a 100 MHz master clock with a serial USB (universal serial bus) interface.

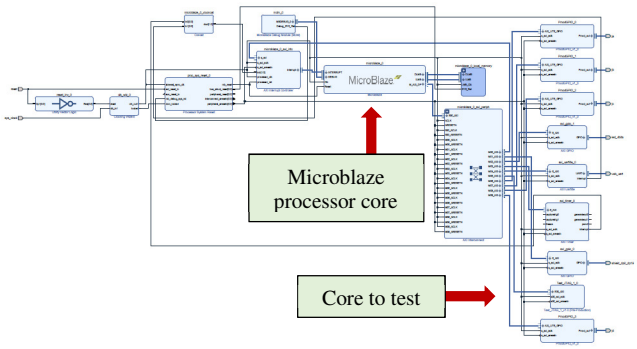


Fig. 4. MicroBlaze block design in Vivado 2019.2

The *CUT* was created as a VHDL (VHSIC (Very High Speed Integrated Circuit) Hardware Description Language (HDL)) [16] module with an AXI interface as shown in Fig. 5. The interface `S00_AXI` provides the necessary data, address, and control signals. The other two connections are the input clock (`S00_axi_aclk`) and active-low reset (`S00_axi_resetn`). The peripheral is accessed by writing 32-bit data to the address in memory that the core is located at. The value written to the core in C as follows:

```
*(baseaddr_p) = ( TRSTn ) + ( TMS * 2 ) + ( TCK * 4 ) + ( TDI * 8 ) + ( a * 16 ) + ( b * 256 ) + ( cin * 4096 );
```

`baseaddr_p` is the memory address and the data value written to this address is a number where the individual signals (`TRSTn`, `TMS`, `TCK`, `TDI` and `cin`) are `0` or `1`. The numbers `a` and `b` are integer numbers in the range 0_{10} to 15_{10}.

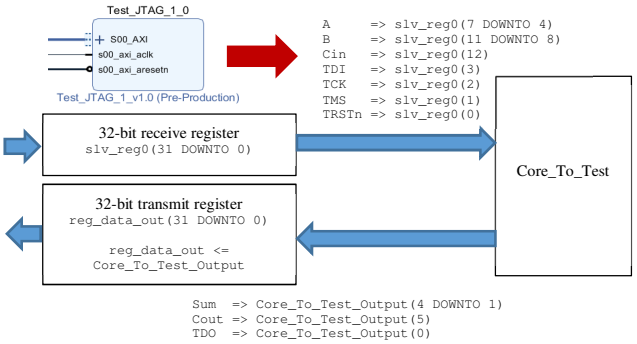
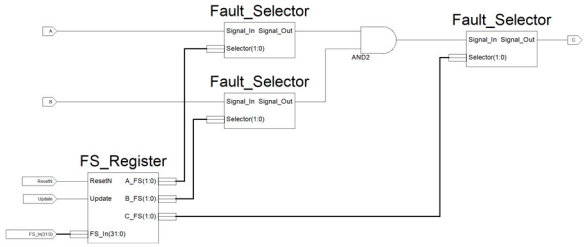


Fig. 5. Circuit (core) under test (`Core_To_Test`) MicroBlaze peripheral

The overall laboratory experiment would be used by connecting the FPGA to an external computer. The learner would load the test set (the test set consisting of the required number of test patterns (a pattern consisting of a test vector and expected response)) into the FPGA using a pre-set protocol and monitor the returned response. This mirrors the same basic approach that is used in simulation, except here rather than a simulation model being used, the hardware within the FPGA emulates a complete IC being tested. The FPGA also emulates the external tester as well as the IC being tested. An advantage of this type of system is that the same hardware design can be accessible to multiple users, the design is readily modified, an instructor can control the operation of the experiment, and specific faults can be introduced into the system whilst operating (e.g., [17]). To include the ability to model a faulty *CUT*, the SAF was introduced by introducing a hardware model of a node operation (i.e., fault-free, stuck-at logic 0, and stuck-at logic 1) into the circuit description before synthesis. This principle is shown in Fig. 6 with reference to a 2-input AND gate. The node SAF status can be selected by loading a code into a register, and this code can be set by the instructor separate to any learner operation.

Modelling node behavior in a logic circuit by inserting faults




```
Selector_Process : PROCESS (Signal_In, Selector)
BEGIN
    CASE (Selector) IS
        WHEN "00" =>    Signal_Out <= Signal_In;
        WHEN "01" =>    Signal_Out <= '0';
        WHEN "10" =>    Signal_Out <= '1';
        WHEN "11" =>    Signal_Out <= Signal_In;
        WHEN OTHERS =>  Signal_Out <= Signal_In;
    END CASE;
END PROCESS Selector_Process;
```

Stuck-at fault selector using a VHDL case statement

Fig. 6. Modelling the stuck-at fault in VHDL as a VHDL case statement

V. Conclusions

This paper has introduced and discussed the design and operation of a flexible FPGA based digital IC test development education laboratory. The set-up was developed to allow for different digital IC test development scenarios to be configured within the FPGA. For analysis and evaluation purposes, embedded machine learning functions could also be incorporated for automated reporting of the system use. A case study, based on the Xilinx Artix-7 FPGA that incorporates a test set-up for an example digital IC, was introduced and its use discussed. The FPGA provides a flexible hardware platform for developing hardware and software implementations of the experiment control arrangement as well as different experiments to undertake.

References

[1] Xilinx, Inc., *Field Programmable Gate Array (FPGA)*, 2020. Accessed on: May 4, 2020. [Online]. Available: https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html

[2] S. L. Hurst, *VLSI Testing, digital and mixed analogue/digital techniques*, IEE Circuits, Devices and Systems Series 9. United Kingdom: The Institution of Electrical Engineers, 1998.

[3] R. Rajsuman, *System-on-a-Chip: Design and Test*. United States of America: Artech House Publishers, 2000.

[4] V. Agrawal and M. Bushnell, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. United States of America: Springer-Verlag NY Inc., 2013.

[5] I. Grout and A. K. bin A'ain, "Introductory Laboratories in Semiconductor Devices using the Digilent Analog Discovery," in *12th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2015.

[6] Xilinx, Inc., *Artix-7*, 2020. Accessed on: May 4, 2020. [Online]. Available: https://www.xilinx.com/products/silicon-devices/fpga/artix-7.html

[7] *Test Access Port and Boundary-Scan Architecture*, IEEE standard 1149.1™-2013 (Revision of IEEE standard 1149.1-2001), 2013.

[8] Xilinx, Inc., *MicroBlaze Soft Processor Core*, 2020. Accessed on: May 4, 2020. [Online]. Available: https://www.xilinx.com/products/design-tools/microblaze.html

[9] Haralampos-G. Stratigopoulos, "Machine learning applications in IC testing," in *23rd European Test Symposium (ETS)*, 2018.

[10] L. Wang and M. Luo, "Machine Learning Applications and Opportunities in IC Design Flow," in *International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2019.

[11] H.-G.D. Stratigopoulos and Y. Makris, "Bridging the accuracy of functional and machine-learning-based mixed-signal testing," in *24th IEEE VLSI Test Symposium*, 2006.

[12] Xilinx, Inc., *Vivado Design Suite*, 2020. Accessed on: May 4, 2020. [Online]. Available: https://www.xilinx.com/products/design-tools/vivado.html

[13] International Organization for Standardization, *ISO/IEC 9899:2018 [ISO/IEC 9899:2018] Information technology — Programming languages — C*, 2020. Accessed on: May 4, 2020. [Online]. Available: https://www.iso.org/standard/74528.html

[14] Xilinx, Inc., *Vitis Platform*, 2020. Accessed on: May 4, 2020. [Online]. Available: https://www.xilinx.com/products/design-tools/vitis/vitis-platform.html

[15] Digilent Inc., *Arty Family*, 2020. Accessed on: May 4, 2020. [Online]. Available: https://store.digilentinc.com/arty

[16] *VHDL Language Reference Manual*, IEEE standard 1076™-2008 (Revision of IEEE standard 1076-2002), 2009.

[17] K. P. Parker, *The Boundary-Scan Handbook, Analog and Digital*, 2nd ed. United States of America: Kluwer Academic Publishers, 2000, pp. 163-165.